

REMARKS

The Examiner has rejected Claims 1-25 under 35 U.S.C. 103(a) as being unpatentable over Sonderegger et al. U.S. Patent No.: 5,859,978 in view of Olds U.S. Patent No.: 5,878,415. Applicant respectfully disagrees with such rejection, especially in view of the amendments made hereinabove. Specifically, applicant has amended each of the independent claims to include the subject matter of Claims 2 and 6 et al. (or similar limitations).

The Examiner has relied on the following excerpts from Sonderegger and Olds to make a prior art showing of applicant's claimed: "management console [that] is in communication with the network directory and the directory server for providing a user interface, the management console being adapted to selectively display the hierarchical tree structure and the control settings stored in the network directory." See all independent claims.

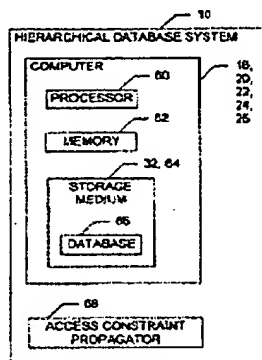


FIG. 3

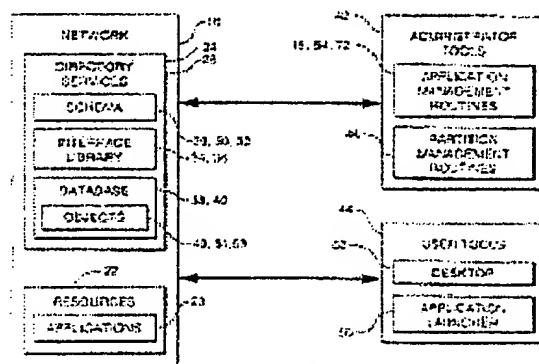


FIG. 2

Such disclosure, however, merely shows the framework or architecture of the prior art systems. There is simply no disclosure, teaching or suggestion of any sort of "management console [that] is in communication with the network directory and the directory server for providing a user interface, the management console being adapted to selectively display the hierarchical tree structure and the control settings stored in the network directory" (emphasis added). Only applicant teaches and claims a user interface, whereby the hierarchical tree structure itself is displayed along with the

display of control settings stored in the network directory. A specific showing of such feature in the prior art or a notice of allowance is respectfully requested.

The Examiner continues by relying on the following excerpts from Sonderegger to make a prior art showing of applicant's claimed: "at least one control setting [that] is a scheduled task and wherein the end node causes performance of the task when the scheduled task is to be performed."

See all independent claims.

"Each application object represents one application program and its execution environment. Thus, in one embodiment each application object includes the location of at least one executable code for the application in question, a brief name which textually identifies the application, an icon which graphically identifies the application, the location of the application's working directory, the drive mappings and printer port captures needed by the application, and the command line parameters (if any) that should be passed to the application when it begins execution. Alternative embodiments of application objects include additional useful information such as a brief textual description ("blurb") describing the application to potential new users, a list of other users who can be contacted to answer questions about the application, licensing information, and scripts to run before and/or after execution of the application.

Collecting such information about each application into one application object in the database assists greatly in preventing the creation or use of inconsistent or obsolete information about the application. Moreover, application objects receive the beneficial characteristics of the database such as hierarchical organization, access control, replication of database objects to provide fault-tolerance, and access through familiar interface tools.

The present invention also provides an application launcher. In one embodiment, the application launcher allows a user to browse through the application objects which represent the applications available to that user and to view the information currently stored in the objects. The launcher also automatically launches specified "auto start" applications when a user runs the launcher. In addition, the launcher updates a list of available applications that is associated with the user's desktop by querying the database. The launcher's update capability relieves administrators of the need to manually update each user's desktop when a new application is added to the network or an established application is moved.

When it is requested to launch an application, the launcher uses the information in the application's database object to setup execution environment resources needed by the application, to then create a process which executes the application, and to finally clean up after the application terminates. Resource setup typically involves mapping drives and capturing printer ports as needed; setup may also involve running a startup script. After the application terminates, the launcher cleans up by unmapping drives, releasing captured ports, and detaching from servers as needed. Cleaning up also includes running a shutdown script if one is provided.

Thus, the present invention provides a computer-implemented method and apparatus for consistently tracking and employing information about application programs and their execution environments in a network. Consistency is achieved by gathering the information into a central database which is accessed through specified administrator tools and user interfaces. Database updates are readily performed with the administrator tools or by the application launcher, thereby reducing the effort associated with changes in the location of executables or in the other information needed to execute an application." (see col. 3, lines 7 - 67)

After a careful review of such excerpt, there appears to simply be no disclosure, teaching or suggestion of any sort of "at least one control setting [that] is a scheduled task and wherein the end node causes performance of the task when the scheduled task is to be performed" (emphasis added). Only applicant teaches and claims specifically a scheduled task in the specific context of a hierarchical tree structure inheritance system and method, as specifically claimed. It is noted that, in the foregoing excerpt from Sonderegger, there is merely a discussion of "access control," which vastly departs from applicant's claimed scheduled task that causes performance of the task when the scheduled task is to be performed. Again, a specific showing of such feature in the prior art or a notice of allowance is respectfully requested.

Still yet, in the spirit of expediting the prosecution of the present application, applicant has amended each of the independent claims to require that "the scheduled task [be] an anti-virus-related task." It is noted that neither Sonderegger nor Olds makes any mention of inheriting scheduled control settings related to security tasks, let alone an anti-virus-related task. A specific showing of such feature in the prior art or a notice of allowance is respectfully requested.

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed.Cir.1991). Applicant respectfully asserts that at least the third element of the *prima facie* case of obviousness has not been met, for the reasons set forth hereinabove.

Applicant further brings to the Examiner's attention the following claim limitations that have been added, and are deemed novel by applicant:

"wherein broad control settings are set higher in the tree while lower level control settings

are set at a level of one of the devices" (see Claim 26)

"wherein the broad control settings require the scanning of all executable files for viruses, cleaning of the files if possible, quarantining of the files if the files can not be cleaned upon detecting a virus, and sending infection reports to a network administrator" (see Claim 27)

"wherein a mid-level control setting is set to report all infections to a local administrator" (see Claim 28)

"wherein the lower level settings require the deletion of infected files of said one of the devices" (see Claim 29)

"a get policy component that obtains relevant policies, a first cache for caching an output of the get policy component, a calculate policy component that calculates an inheritance with the output received from the first cache and transmits the inheritance to a second cache, wherein the information in the first cache is updated based on a time stamping" (see Claim 30)

A specific showing of such features in the prior art or a notice of allowance is respectfully requested.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 505-5100. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 50-1351 (Order No. NAI1P280).

Respectfully submitted,

Kevin J. Zilka
Registration No. 41,429

P.O. Box 721120
San Jose, CA 95172-1120
408-505-5100

Docket: NAI1P280_99.121.01

-14-